PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-117892
(43)Date of publication of application : 27.04.2001

(51)Int.CI. G06F 15/173
G06F 15/177

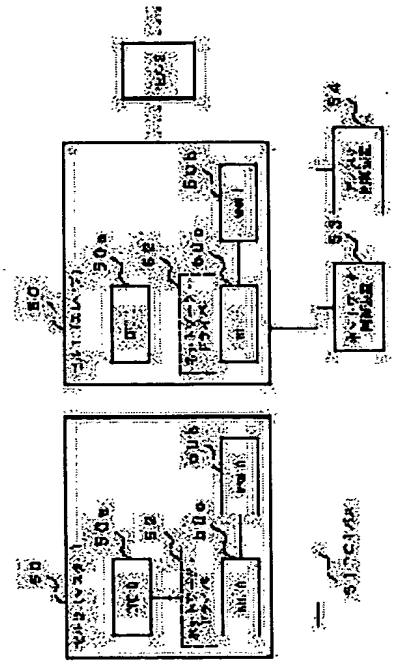
(21)Application number : 2000-030406
(22)Date of filing : 08.02.2000

(71)Applicant : NEC CORP
(72)Inventor : TAKAGI HITOSHI

(30)Priority
Priority number : 1999 417643 Priority date : 14.10.1999 Priority country : US

(54) MULTI-PROCESSOR COMPUTER SYSTEM FOR COMMUNICATION VIA INTERNAL BUS AND ITS COMMUNICATION METHOD

(57)Abstract:
PROBLEM TO BE SOLVED: To provide an environment where processors execute communication via a connecting device through the used of a network protocol.
SOLUTION: A multi-processor computer system is provided with a plurality of computing cells having a processor, a memory and an interface unit and with the connecting device for connecting the plurality of computing cells through the use of the interface unit. The computing cells are respectively provided with a network driver for the connecting device and execute communication via the connecting device by using the network protocol.



LEGAL STATUS

[Date of request for examination] 16.01.2001
[Date of sending the examiner's decision of rejection] 19.11.2002
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision

of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2001－117892
（P2001－117892A）
(43)公開日　平成13年４月27日(2001.4.27)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード（参考）
Ｇ０６Ｆ　15/173		Ｇ０６Ｆ　15/173	Ａ	５Ｂ０４５
15/177	６７６	15/177	６７６Ａ	

審査請求　有　請求項の数15　ＯＬ　（全 13 頁）

(21)出願番号　特願2000－30406(P2000－30406)

(22)出願日　平成12年２月８日(2000.2.8)

(31)優先権主張番号　０９／４１７６４３

(32)優先日　平成11年10月14日(1999.10.14)

(33)優先権主張国　米国（ＵＳ）

(71)出願人　000004237
日本電気株式会社
東京都港区芝五丁目７番１号

(72)発明者　▲高▼木　均
東京都港区芝五丁目７番１号　日本電気株式会社内

(74)代理人　100093595
弁理士　松本　正夫

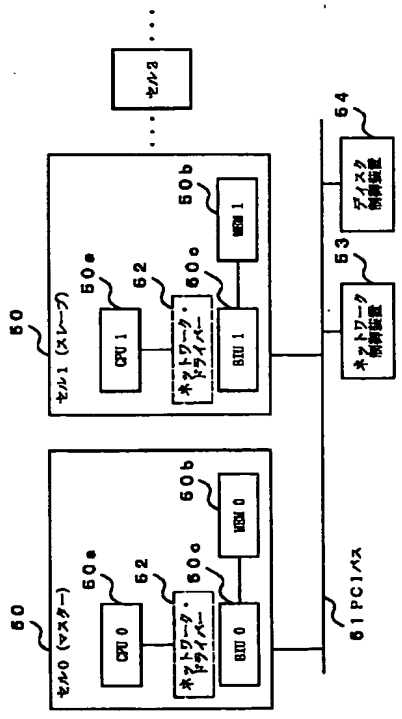
Ｆターム(参考)　5B045 AA03 BB12 BB17 BB28 BB29
BB42 BB47

(54)【発明の名称】　内部バスを介して通信するマルチプロセッサ・コンピュータシステムとその通信方法

(57)【要約】

【課題】　プロセッサがネットワーク・プロトコルを使用し、相互接続装置を介して互いに通信する環境を提案する。

【解決手段】　マルチプロセッサ・コンピュータシステムであって、プロセッサ、メモリ、及びインターフェース・ユニットを備える複数のコンピューティング・セルと、インターフェース・ユニットを使用して複数の前記コンピューティング・セルを相互接続する相互接続装置とを備え、コンピューティング・セルは、相互接続装置用のネットワーク・ドライバを備え、ネットワーク・プロトコルを使用することにより相互接続装置を介して互いに通信する。



【特許請求の範囲】

【請求項１】　マルチプロセッサ・コンピュータシステムであって、下記要件を備える。プロセッサ、メモリ、及びインターフェース・ユニットを備える複数のコンピューティング・セルと、

インターフェース・ユニットを使用して複数の前記コンピューティング・セルを相互接続する相互接続装置とを備え、

前記コンピューティング・セルは、前記相互接続装置用のネットワーク・ドライバを備え、ネットワーク・プロトコルを使用することにより前記相互接続装置を介して互いに通信する。

【請求項２】　前記相互接続装置が、業界標準の共用入出力バスであることを特徴とする請求項１に記載のマルチプロセッサ・コンピュータシステム。

【請求項３】　前記共用入出力バスが、ＰＣＩバスであることを特徴とする請求項２に記載のマルチプロセッサ・コンピュータシステム。

【請求項４】　前記相互接続装置が、交換相互接続装置であることを特徴とする請求項１に記載のマルチプロセッサ・コンピュータシステム。

【請求項５】　前記ネットワーク・プロトコルが、ＴＣＰ／ＩＰプロトコルであることを特徴とする請求項１に記載のマルチプロセッサ・コンピュータシステム。

【請求項６】　マルチプロセッサ・コンピュータシステムのプロセッサ間で通信する通信方法であって、下記ステップを備える。マルチプロセッサ・コンピュータシステムのプロセッサ、メモリ、及びインターフェース・ユニットを備える複数のコンピューティング・セルを提供し、

相互接続装置を使用して前記コンピューティング・セルを接続し、

前記コンピューティング・セルに、相互接続装置用ネットワーク・ドライバを提供し、

前記相互接続装置用ネットワーク・ドライバにより、前記コンピューティング・セルがネットワーク・プロトコルを使用することにより前記相互接続装置を介して互いに通信する。

【請求項７】　前記相互接続装置として、業界標準共用入出力バスを提供することを特徴とする請求項６に記載の通信方法。

【請求項８】　前記共用入出力バスが、ＰＣＩバスであることを特徴とする請求項７に記載の通信方法。

【請求項９】　前記相互接続装置が、交換相互接続装置であることを特徴とする請求項６に記載の通信方法。

【請求項１０】　前記ネットワーク・プロトコルが、ＴＣＰ／ＩＰプロトコルであることを特徴とする請求項６に記載の通信方法。

【請求項１１】　マルチプロセッサ・コンピュータシステムのプロセッサ間で通信する通信方法であって、下記

ステップを備える。少なくともマルチプロセッサ・コンピュータシステムのプロセッサ、メモリ、及びインターフェース・ユニットを備える複数のコンピューティング・セルを提供し、

相互接続装置を使用して前記コンピューティング・セルを接続し、

前記コンピューティング・セルのプロセッサ間で前記相互接続装置を介してデータを送信するためにネットワーク・プロトコルを使用する。

【請求項１２】　前記相互接続装置として共用入出力バスを提供するステップをさらに備えることを特徴とする請求項１１に記載の通信方法。

【請求項１３】　前記共用Ｉ／ＯとしてＰＣＩバスを提供するステップをさらに備えることを特徴とする請求項１２に記載の通信方法。

【請求項１４】　前記相互接続装置として、交換相互接続装置を提供するステップをさらに備えることを特徴とする請求項１１に記載の通信方法。

【請求項１５】　前記ネットワーク・プロトコルとしてＴＣＰ／ＩＰプロトコルを使用するステップをさらに備えることを特徴とする請求項１１に記載の通信方法。

【発明の詳細な説明】

【０００１】

【発明が属する技術分野】本発明は、マルチプロセッサ・コンピュータシステムの分野に関し、特に、ネットワーク・プロトコルを使用し、内部バスその他の相互接続装置を介してプロセッサが互いに通信するマルチプロセッサ・コンピュータシステムに関する。

【０００２】

【従来の技術】ネットワーク・コンピューティング環境においては、ユーザー・アプリケーション・プログラムはネットワークのアプリケーション層プロトコルを使用して通信する。例えば、デスクトップ・システム（クライアント・マシン）上の電子メール・プログラムは、ＰＯＰ（Ｐｏｓｔ　Ｏｆｆｉｃｅ　Ｐｒｏｔｏｃｏｌ）やＳＭＴＰ（Ｓｉｍｐｌｅ　Ｍａｉｌ　Ｔｒａｎｓｆｅｒ　Ｐｒｏｔｏｃｏｌ：メール転送プロトコル）などのメール転送プロトコルを使用してサーバ・コンピュータ上のメール・サーバ・プログラムと通信する。ネットワーク接続されたアプリケーション・システムの他の例では、ウェブ・ブラウザーはＨＴＴＰ（Ｈｙｐｅｒ　Ｔｅｘｔ　Ｔｒａｎｓｆｅｒ　Ｐｒｏｔｏｃｏｌ）を利用してウェブ・サーバと通信する。

【０００３】アプリケーション・プログラムは、ＴＣＰ／ＩＰシステムに取り付けたＳｏｃｋｅｔｓ　ＡＰＩ、ＮｅｔＢＩＯＳ、ＴＬＩ（Ｔｒａｎｓｐｏｒｔ　ＬａｙｅｒＩｎｔｅｒｆａｃｅ）ＡＰＩなど、何らかのＡＰＩ（ＡｐｐｌｉｃａｔｉｏｎＰｒｏｇｒａｍ　Ｉｎｔｅｒｆａｃｅ：アプリケーション・プログラム・インターフェース）を介してネットワークと通信する。ＡＰＩの下

には、プログラム間でデータをやり取りする際に、データの喪失その他のエラーが発生することなく正常に転送されるようにする各種プロトコル・スタックの層がある。こうしたプロトコル・スタックの例としては、ＴＣＰ／ＩＰ、ＳＰＸ／ＩＰＸ、ＮｅｔＢＥＵＩなどが挙げられる。これらのプロトコルはそれぞれ、ネットワーク・ドライバと呼ばれるプログラムによって実装される。

【０００４】ネットワーク・ドライバの下には、ネットワーク接続された２点間でデータ転送を行うために適切なハードウェアを実際に制御するハードウェア・ドライバがある。これらのハードウェア・ドライバとそれに対応するハードウェアは、イーサネット（ＩＥＥＥ　８０２．３）、ＡＴＭ（Ａｓｙｎｃｈｒｏｎｏｕｓ　Ｔｒａｎｓｆｅｒ　Ｍｏｄｅ：非同期転送モード）、トークン・リング（ＩＥＥＥ８０２．５）などの標準仕様や、ＩＳＤＮ（Ｉｎｔｅｇｒａｔｅｄ　Ｓｕｂｓｃｒｉｂｅｒ　Ｄｉｇｉｔａｌ　Ｎｅｔｗｏｒｋ：統合ディジタル通信サービス網）、Ｘ．２５、フレーム・リレー、ＤＳＬ（Ｄｉｇｉｔａｌ　Ｓｕｂｓｃｒｉｂｅｒ　Ｌｉｎｅ）、ＡＤＳＬ（Ａｓｙｎｃｈｒｏｎｏｕｓ　Ｄｉｇｉｔａｌ　Ｓｕｂｓｃｒｉｂｅｒ　Ｌｉｎｅｓ）などのその他のＷＡＮ（Ｗｉｄｅ　ＡｒｅａＮｅｔｗｏｒｋｉｎｇ：広域ネットワーク）標準を使用して実装される。最後に、コンピュータは、一般にこれらの標準の少なくとも１つに準拠するケーブルによって物理的に接続される。

【０００５】一般に、ネットワーク・インターフェース・ハードウェアは、コンピュータの内部に設けられる。その例としては、イーサネット・カードなどのイーサネット・ハードウェア・コンポーネントをコンピュータのＩ／Ｏ（Ｉｎｐｕｔ／Ｏｕｔｐｕｔ：入出力）バスに接続した構成が考えられる。このようなイーサネット・カードは、Ｉｎｔｅｌ　Ｃｏｒｐｏｒａｔｉｏｎなどのベンダーによって製造されている。あるいは、２枚のイーサネット・カードをＩ／Ｏバスに取り付ける場合であれば、一方のカードをＩｎｔｅｌ製にし、もう一方のカードをＣｏｍｐａｑ　Ｃｏｒｐｏｒａｔｉｏｎなどの他のベンダーが製造したものとすることもできる。後者の場合、ネットワーク・ドライバの最下層が、コンピュータの構成とアプリケーション・プログラムから発行された要求を参照して、使用できるハードウェア・カードを特定する。

【０００６】図１０及び図１１は、分散アプリケーション・システム内のネットワーキング層の例である。図１０は、Ｓｏｃｋｅｔｓ、ＴＬＩ、ＣＰＩ／Ｃ、ＮａｍｅｄＰｉｐｅｓ　ＡＰＩなどのトランスポート非依存ＡＰＩ１１のサービスを起動するアプリケーション・プログラム１０を示す。これらのＡＰＩは、ＴＣＰ／ＩＰ、ＳＮＡ、ＩＰＸ／ＳＰＸ、ＮｅｔＢＩＯＳを始めとする各種ネットワーキング転送プロトコル１２の実装へのアクセスを提供する。転送プロトコル１２の実装の下には、ＮＤＩＳやＯＤＩインターフェースなどの標準インターフェースによってアクセス可能な論理ネットワーク・ドライバ１３の層がある。これらの論理ネットワーク・ドライバは、トークン・リング、イーサネット、ＩＳＤＮ仕様などの標準仕様を使って実装されたハードウェア１４と協働して動作する。

【０００７】図１１は、最も一般的に使用されるネットワーキング・ソフトウェアの層が、最上層にアプリケーション・プログラムのインターフェースを提供するアプリケーション層１５を備えるＯＳＩ参照モデルの７つの論理層にどのようにマッピングされるかを示したものである。最下層の物理層２１は、ネットワーキング・ハードウェアと媒体に関する仕様を定義する。ＯＳＩモデルの中間層には、プレゼンテーション層１６、セッション層１７、トランスポート層１８、ネットワーク層１９及び論理リンク層２０が含まれる。

【０００８】インターネット・ウェブ・ブラウザー（Ｉｎｔｅｒｎｅｔ　ＥｘｐｌｏｒｅｒやＮｅｔｓｃａｐｅ　Ｎａｖｉｇａｔｏｒなど）とインターネット・ウェブ・サーバとの通信の基礎をなす層について論じるため、その一例を図示する。図１２に示すように、インターネット・ウェブ・ブラウザー２２は、ＨＴＴＰアプリケーション層プロトコルを使ってインターネット・ウェブ・サーバと通信する。ブラウザーがＨＴＴＰによって通信するためには、ＴＣＰ／ＩＰプロトコルなどの下位レベルのプロトコルを使用して、すべてのデータがエラーやデータの喪失が発生することなく正常に送信されるようにする必要がある。

【０００９】ＴＣＰ／ＩＰ実装は、ＴＣＰ／ＩＰソフトウェアのサービスを起動するためのＳｏｃｋｅｔｓ　ＡＰＩと呼ばれるＡＰＩ　２３を提供する。したがって、このＡＰＩは、アプリケーション・プログラムがネットワーク・ドライバ２４のサービスを起動するための一手段となる。ネットワーク・ドライバ・ソフトウェア２４は、ＴＣＰ／ＩＰプロトコルを実装するために必要である。ネットワーク・ドライバ２４は、ネットワークからの情報を転送、受信、または制御するためのｒｅａｄ、ｗｒｉｔｅ、及びｉｏｃｔｌと呼ばれるライブラリー・ルーチンを呼び出すことによってハードウェア・ドライバ２５を起動する。ネットワーク・ドライバ２４には、オペレーティング・システムによって、コンピュータ内にイーサネット・アダプターまたはＡＴＭアダプターが備えられているかどうかといったネットワーキング・ハードウェア２６の構成情報が供給される。

【００１０】そのため、マシン間のデータ転送における主要なオペレーションには、基本「送信」オペレーションが含まれる。したがって、この機能を起動するためのハードウェア・ドライバの基本オペレーションは、ネットワーキング・ハードウェアにネットワークを介して供

給されたデータを送信させるためにハードウェア・ドライバから供給されるＳＥＮＤコマンド（または、関数呼び出し）である。同様に、ＲＥＡＤの基本オペレーションでは、ハードウェア・ドライバがネットワークから送信されたデータを読み取る。

【００１１】「Ｃｏｍｐｕｔｅｒ　Ａｒｃｈｉｔｅｃｔｕｒｅ：　Ａ　Ｑｕａｎｔｉｔａｔｉｖｅ　ａｐｐｒｏａｃｈ（コンピュータ・アーキテクチャー：　定量的アプローチ）」、Ｊｏｈｎ　Ｈｅｎｎｅｓｓｙ　ａｎｄ　Ｄａｖｉｄ　Ｐａｔｔｅｒｓｏｎ、第２版、Ｍｏｒｇａｎ　Ｋａｕｆｍａｎ、１９９６という題名のテキストには、従来のマルチプロセッサ・コンピュータシステムの例が示されている。このテキストに示される最初の例は、並行処理コンピュータシステムのノード群３１を接続する汎用相互接続ネットワーク３０を示す図１３を参照して説明されている。例えば、ＭＰＰ（Ｍａｓｓｉｖｅｌｙ　Ｐａｒａｌｌｅｌ　Ｐｒｏｃｅｓｓｏｒ）は、最大距離が非常に短い（２５メートル未満のことも多い）何千ものノード群３１を相互接続できる。

【００１２】このテキストの２番目の例について説明する図１４を参照すると、分散メモリ装置が示されている。この分散メモリ装置は、個別にプロセッサ、何らかのメモリ３４、典型的には何らかのＩ／Ｏ　３５、及び相互接続装置３２へのインターフェースを備えるノード群３３で構成される。

【００１３】これとは対照的に、数台のコンピュータシステムを相互に接続する従来のコンピュータ・ネットワークでは、コンピュータシステムはネットワーク・プロトコルを使用して、ＮＩＣ（Ｎｅｔｗｏｒｋ　Ｉｎｔｅｒｆａｃｅ　Ｃａｒｄ）などのネットワーク・ハードウェアとネットワーク・ドライバを備えるネットワーク・ソフトウェアを介してネットワークと通信する。典型的には、コンピュータ・ネットワークのノード間の通信を調整するために、ネットワーク・オペレーティング・システム（ＮＯＳ）が備えられる。

【００１４】

【発明が解決しようとする課題】しかし、これまで開示された従来技術のシステムでは、内部またはＣＰＵバス、または類似の相互接続装置を別個のネットワーキング装置として扱うことにより、マルチプロセッサ・コンピュータシステム内の異なるプロセッサがネットワーク・プロトコルを使用して互いに通信できるようにするものは存在しない。すなわち、上記の従来技術ではいずれも、複数のプロセッサが共用バスなどの適切な相互接続装置によって互いに接続されたマルチプロセッサ・コンピュータシステム内のプロセッサ間でネットワーク・プロトコルを使用する技術は開示されていない。したがって、上記の従来技術ではいずれも、マルチプロセッサ・コンピュータシステム内のプロセッサが標準ネットワーク・プロトコルを使用し、相互接続装置を介して互いに通信できるようにするために、相互接続装置用ネットワーク・ドライバが提案されていない。

【００１５】本発明の１つの目的は、プロセッサがネットワーク・プロトコルを使用し、相互接続装置を介して互いに通信するマルチプロセッサ・コンピュータシステムを提供することにある。

【００１６】本発明の他の目的は、プロセッサがネットワーク・プロトコルを使用し、共用バスを介して互いに通信するマルチプロセッサ・コンピュータシステムを提供することにある。

【００１７】本発明の他の目的は、プロセッサがＴＣＰ／ＩＰプロトコルを使用し、共用ＰＣＩバスを介して互いに通信するマルチプロセッサ・コンピュータシステムを提供することにある。

【００１８】本発明の他の目的は、プロセッサがネットワーク・プロトコルを使用し、交換相互接続装置を介して互いに通信するマルチプロセッサ・コンピュータシステムを提供することにある。

【００１９】本発明の他の目的は、プロセッサがＴＣＰ／ＩＰプロトコルを使用し、交換相互接続装置を介して互いに通信するマルチプロセッサ・コンピュータシステムを提供することにある。

【００２０】本発明の他の目的は、プロセッサがネットワーク・プロトコルを使用し、相互接続装置を介して通信するマルチプロセッサ・システムのプロセッサ間で通信する通信方法を提供することにある。

【００２１】本発明のさらなる目的は、プロセッサがネットワーク・プロトコルを使用し、共用バスを介して通信するマルチプロセッサ・システムのプロセッサ間で通信する通信方法を提供することにある。

【００２２】本発明の他の目的は、プロセッサがネットワーク・プロトコルを使用し、交換相互接続装置を介して通信するマルチプロセッサ・システムのプロセッサ間で通信する通信方法を提供することにある。

【００２３】本発明のさらなる目的は、プロセッサがＴＣＰ／ＩＰネットワーク・プロトコルを使用し、相互接続装置を介して通信するマルチプロセッサ・システムのプロセッサ間で通信する通信方法を提供することにある。

【００２４】

【課題を解決するための手段】上記目的を達成するため、本発明によれば、マルチプロセッサ・コンピュータシステムであって、プロセッサ、メモリ、及びインターフェース・ユニットを備える複数のコンピューティング・セルと、インターフェース・ユニットを使用して複数の前記コンピューティング・セルを相互接続する相互接続装置とを備え、前記コンピューティング・セルは、前記相互接続装置用のネットワーク・ドライバを備え、ネットワーク・プロトコルを使用することにより前記相互接続装置を介して互いに通信する。

【００２５】請求項２の本発明によれば、前記相互接続装置が、業界標準の共用入出力バスであることを特徴とする。

【００２６】請求項３の本発明によれば、前記共用入出力バスが、ＰＣＩバスであることを特徴とする。

【００２７】請求項４の本発明によれば、前記相互接続装置が、交換相互接続装置であることを特徴とする。

【００２８】請求項５の本発明によれば、前記ネットワーク・プロトコルが、ＴＣＰ／ＩＰプロトコルであることを特徴とする。

【００２９】請求項６の本発明によれば、マルチプロセッサ・コンピュータシステムのプロセッサ間で通信する通信方法であって、マルチプロセッサ・コンピュータシステムのプロセッサ、メモリ、及びインターフェース・ユニットを備える複数のコンピューティング・セルを提供し、相互接続装置を使用して前記コンピューティング・セルを接続すし、前記コンピューティング・セルに、相互接続装置用ネットワーク・ドライバを提供し、前記相互接続装置用ネットワーク・ドライバにより、前記コンピューティング・セルがネットワーク・プロトコルを使用することにより前記相互接続装置を介して互いに通信する。

【００３０】請求項１１の本発明によれば、マルチプロセッサ・コンピュータシステムのプロセッサ間で通信する通信方法であって、少なくともマルチプロセッサ・コンピュータシステムのプロセッサ、メモリ、及びインターフェース・ユニットを備える複数のコンピューティング・セルを提供し、相互接続装置を使用して前記コンピューティング・セルを接続し、前記コンピューティング・セルのプロセッサ間で前記相互接続装置を介してデータを送信するためにネットワーク・プロトコルを使用する。

【００３１】

【発明の実施の形態】本発明は、一般的な態様では、複数のコンピューティング・セルを備え、各セルがプロセッサ、メモリ、及びネットワーク・インターフェース・ユニットからなるマルチプロセッサ・コンピュータシステムを提供する。相互接続装置はコンピューティング・セルを接続する。コンピューティング・セルには、ＴＣＰ／ＩＰネットワーク・プロトコルなどのネットワーク・プロトコルを使用し、相互接続装置を介して互いに通信できるようにするための相互接続装置用ネットワーク・ドライバが設けられる。

【００３２】他の一般的な態様では、本発明は、複数個のコンピューティング・セルを提供し、各コンピューティング・セルがプロセッサ、メモリ、及びネットワーク・インターフェース・ユニットを備えるマルチプロセッサ・システムのプロセッサ間で通信するための方法を提供する。接続ステップは、相互接続装置を提供することによってコンピューティング・セルを互いに接続する。さらなる提供ステップは、各コンピューティング・セルに、ＴＣＰ／ＩＰネットワーク・プロトコルなどのネットワーク・プロトコルを使用し、相互接続装置を介して互いに通信できるようにするための相互接続装置用ネットワーク・ドライバを提供する。

【００３３】したがって本発明は、内部バスが、メッセージがプロセッサ間で移動できるネットワーク・ハードウェアとして構成され定義される構成を提供する。ネットワーク・ドライバの最下層は、イーサネット・ハードウェア・カードなどのネットワーク・ハードウェアと類似した方法で、内部バスまたは類似のネットワーク装置としての相互接続装置を認識し通信できるようにプログラムされる。

【００３４】ネットワーク・ドライバを内部バスまたは類似の相互接続装置と通信できるようにプログラムすることは、当該技術に精通した当業者が通常有する能力である。例えば、リナックス環境でネットワーク・ドライバをプログラムする方法の詳細は、インターネット上のＵＲＬ“ｈｔｔｐ：／／ｗｗｗ．ｌｉｎｕｘｄｏｃ．ｏｒｇ／ＨＯＷＴＯ／Ｅｔｈｅｒｎｅｔ－ＨＯＷＴＯ－８．ｈｔｍｌ＃ｓｓ８．２”で提供されている。

【００３５】例えば、ＴＣＰ／ＩＰ実装におけるこのようなネットワーク・ドライバの実装は、ＩＰパケットが形成されてから、ネットワーク・インターフェース・ハードウェアを介してネットワークに送信されるまでの間にそれを傍受するようにプログラムされたソフトウェアとすることができる。この特別プログラム式ソフトウェアは、ＩＰアドレスがマルチプロセッサ・システムの他のプロセッサの１つを指定しているかどうか判断し、指定している場合は、該当するプロセッサを宛先とし内部バスまたはその他の相互接続装置を介してパケットをルーティングする。ＩＰアドレスが他のプロセッサの１つに属さないと判断した場合、特別プログラム式ソフトウェアは、さらなる転送のため通常の方法で、ネットワークを宛先とし、ネットワーク・インターフェース・ハードウェアを介してＩＰパケットをルーティングする。以下に、当該ソフトウェアがＩＰパケットをプロセッサ間でルーティングするための手法の例を取り上げて詳細に説明する。ここでは、マルチプロセッサＷｉｎｄｏｗｓＮＴ環境の例として、タスク間メッセージングを使用して異なるプロセッサ間のデータ転送を行う場合を使用することができる。

【００３６】図１は、本発明の好適な実施の形態によるシステム構成を示す図である。このシステムには数個のコンピューティング・セル５０が備えられ、各セルはＣＰＵやメモリなどの従来のコンピュータ・コンポーネントで構成されている。各セルは業界標準Ｉ／ＯバスであるＰＣＩバス５１に取り付けられている。各セルは、ネットワーク・プロトコルを使用し、共用バス５１を介して互いに通信する。バス上で通信するためのネットワー

ク・プロトコルとしては、例えば、ＴＣＰ／ＩＰプロトコルを使用することができる。

【００３７】図１に示すように、このマルチプロセッサ・システムは、ＰＣＩバス５１に取り付けられた２以上のコンピューティング・セル５０を備える。各セル５０は、ＣＰＵ　５０ａ、メモリ装置（ＭＥＭ）５０ｂ、及びＢＩＵ（Ｂｕｓ　Ｉｎｔｅｒｆａｃｅ　Ｕｎｉｔ）５０ｃを備える。セルの１つであるセル（０）は、マスター・セル５０（マスター）と呼ばれ、その他のセルはスレーブ・セル５０（スレーブ）と呼ばれる。各ＢＩＵ５０ｃは、ＰＣＩバス５１へのインターフェースを備える。マスター・セルのＢＩＵ５０ｃ（ＢＩＵ０）は、通常のＰＣＩバス・インターフェースＬＳＩである。このようなＬＳＩの例としては、ＩｎｔｅｌＣｏｒｐｏｒａｔｉｏｎの４３０ＴＸ　ＬＳＩがある。このＬＳＩの詳細は、Ｉｎｔｅｌ社のいくつかの刊行物に示されているほか、“ｈｔｔｐ：／／ｄｅｖｅｌｏｐｅｒ．ｉｎｔｅｌ．ｃｏｍ／ｄｅｓｉｇｎ／ｃｈｉｐｓｅｔｓ／ｍａｔｕｒｅ／４３０ｔｘ／ｉｎｄｅｘ．ｈｔｍ”のアドレス（ＵＲＬ）のウェブ・サイトでも入手できる。

【００３８】各セル５０は、さらにＣＰＵバス（ＰＣＩバスなど）５１をネットワーク・デバイスとして認識し、それと通信するプログラム式ネットワーク・ドライバ５２も備えているので、ＴＣＰ／ＩＰネットワーク・プロトコルなどの標準ネットワーク・プロトコルを実装するネットワーキング・ソフトウェアは当該ＣＰＵバス５１を通信媒体として使用できる。

【００３９】スレーブ・セルのＢＩＵ５０ｃ（ＢＵＩ１など）も従来のＰＣＩバスインターフェースＬＳＩに類似しているが、以下のメモリ・エイリアス機能とＤＭＡ（Ｄｉｒｅｃｔ　Ｍｅｍｏｒｙ　Ａｃｃｅｓｓ：直接メモリ・アクセス）機能が追加されている点が異なる。ＣＰＵは、少なくとも３２ビット・データ・パス及び３２ビット・アドレス指定能力を有するのが望ましい。

【００４０】ＰＣＩバス５１には、さらに追加のハードウェアを取り付けることもできる。この追加のハードウェアとしては、例えば、マスター・セルがディスク・ドライブにアクセスするためのディスク制御装置５４や、マスター・セルが他のシステムもしくはネットワークと通信するためのイーサネット制御装置などのネットワーク制御装置５３が挙げられる。

【００４１】本実施の形態においては、自セルのメモリは「ローカル・メモリ」として参照される。他セル内のメモリは、マスター・セル内のメモリを含み、「リモート・メモリ」として参照される。

【００４２】図２は、サブセル１５０がＣＰＵ１５０ａとキャッシュ１５５を備える代替の実施の形態を示す。各サブセル１５０はＣＰＵバス１５６に接続されており、ＣＰＵバス１５６は共用メモリ装置１５７にも接続されている。共用メモリ装置１５７は、共用メモリ１５７ｂとメモリ制御装置１５７ａを備えている。ＣＰＵバス１５６は、ＰＣＩ制御装置１５１ａを介してＰＣＩバス１５１と相互接続されている。サブセル１５０と共用メモリ１５７のこの構成（図２内の点線部分）は、図１に示す実施の形態のセル５０に対応する。したがって、サブセル１５０のこれらのグループは、図１のセル５０がＰＣＩバス５１を使用して互いに接続されているのと同様に、ＰＣＩバス１５１を使用して互いに接続できる。同様に、ＰＣＩバス１５１は、例えばネットワーク制御装置１５３及びディスク制御装置１５４などを介して外部装置と接続できる。この構成では、プログラム式ネットワーク・ドライバ・ソフトウェア１５２は、プロセッサ１５０ａがネットワーク・プロトコルを使用しＣＰＵバス１５６を介して同じグループのサブセル内にある他のプロセッサと通信できるようにプログラムすることが可能である。さらに、プロセッサ１５０ａは、サブセル１５０の他のグループに属するプロセッサ１５０ａと通信することもできる。

【００４３】上述したように、この特別プログラム式ネットワーク・ドライバは、当業者であれば通常有する能力の範囲内で数通りにプログラムできる。こうした実施例では、このプログラム式ドライバ・ソフトウェアをＴＣＰ／ＩＰ実装内でＩＰパケットを傍受するように実施して、他のプロセッサの１つを宛先としてアドレス指定されたＩＰパケットが、ネットワーク・インターフェース・ハードウェアを介してネットワークに送出されるのではなく、共用ＣＰＵバス１５６及び／またはＰＣＩバス１５１上を送信されるようにすることが可能である。

【００４４】図３～図８は、共用バスか類似の相互接続装置によって接続された、図１で開示されるセル間のデータ転送の例を示す概略図である。この概略図のうち、図３はマスターＢＩＵによって実行される機能を示すフローチャートである。マスターＢＩＵは、自セルＣＰＵとＰＣＩバスの両方から送られてくる要求を管理する。

【００４５】ステップ６０５では、マスターＢＩＵは、受信した要求がＣＰＵ要求かどうか判断し、ＣＰＵ要求の場合は、ステップ６１０に進む。ステップ６１０では、マスターＢＩＵはそのＣＰＵ要求のアドレスが最大設置メモリ５０ｂより小さいかどうか判断し、小さい場合は、ステップ６１５でその要求の行き先はマスター・セルのメモリとされる。一方、ＣＰＵ要求のアドレスがステップ６１０の最大設置メモリより大きい場合は、ステップ６２０でその要求の行き先はＰＣＩバスとなる。

【００４６】ステップ６０５で要求がＣＰＵ要求でないと判断した場合、ＢＩＵはステップ６２５に進み、その要求がＰＣＩ要求かどうか判断する。ＰＣＩ要求と判断した場合、ＢＩＵはステップ６３０に進み、ＰＣＩ要求のアドレスが最大設置メモリより小さいかどうか判断する。小さい場合は、ステップ６４０でその要求の行き先はマスター・セルのメモリ５０ｂとなる。そうでない場

合、すなわちＣＰＵ要求のアドレスが最大設置メモリより大きい場合、ステップ６３５でその要求の行き先はマスター・セルの内部レジスタとなる。

【００４７】図４は、スレーブＢＩＵによって実行される機能を示すフローチャートである。スレーブＢＩＵは，スレーブＣＰＵ、ＢＩＵ内のＤＭＡモジュール、及びＰＣＩバスから送られてくる要求を管理する。ステップ７０５で、ＢＩＵは要求がＣＰＵ要求かどうか判断する。ＣＰＵ要求と判断した場合、ＢＩＵはステップ７１０で、要求されたアドレスがスレーブ・セル内の最大設置メモリより小さいかどうか判断する。要求のアドレスが最大設置メモリより小さい場合は、ステップ７１５でその要求がスレーブ・セル内のローカル・メモリに向けられ、そうでない場合、ステップ７２０でその要求の行き先はＰＣＩバスとなる。

【００４８】ステップ７０５で要求がＣＰＵ要求でないと判断した場合、ＢＩＵはステップ７２５に進み、その要求がＰＣＩ要求かどうか判断する。ＰＣＩ要求の場合、ＢＩＵはステップ７３０に進み、ＰＣＩ要求のアドレスが最大設置メモリより小さいかどうか判断する。小さい場合は、ステップ７３５でその要求の行き先はローカル・メモリとなる。そうでない場合、すなわちＰＣＩ要求のアドレスが最大設置メモリより大きい場合、ステップ７４０でその要求の行き先はスレーブ・セルの内部レジスタとなる。

【００４９】図５は、本発明のマルチプロセッサ・システム内で実行されるメモリ・エイリアス機能のステップを示すフローチャートである。前述したように、ＣＰＵ要求の行き先はローカル・メモリかＰＣＩバスのいずれかとなる。要求のアドレスが設置メモリの最高位アドレスより低い場合、その要求の行き先はローカル・メモリにされる。要求のアドレスが設置メモリの最高位アドレスより高い場合、その要求の行き先はＰＣＩバスとなる。ＰＣＩバスのアドレス・スペースの最下部は、マスター・セル（図１ではマスターセル５０（セル０））のメモリ・スペースに割り当てられる。

【００５０】システム内の他のスレーブ・セルに備えられるメモリにアクセスできるようにするため、「エイリアス」手法も採用されている。「エイリアス」は、ＰＣＩバスのアドレス・スペースにアクセスすることによって各セルが他のセルのメモリにアクセスできるようにする機能である。ＰＣＩバス内の事前定義されたアドレスが、セル内のメモリにマッピングされる。各セルは、ＰＣＩバスのアドレス・スペース内に異なる基底アドレスを有する。例えば、図１を参照すると、ＰＣＩアドレス・スペース内のアドレス「０ｘａ０００００００」（０ｘは１６進数を示す）は、セル０のメモリのアドレス「０ｘ００００００００」にマッピングされ、ＰＣＩバスのアドレス・スペース内のアドレス「０ｘａ１０００００００」は、セル１のメモリのアドレス「０ｘ００００００００」にマッピングされ、ＰＣＩバスのアドレス・スペース内のアドレス「０ｘａ２０００００００」は、セル２のメモリのアドレス「０ｘ００００００００」にマッピングされる。

【００５１】各ＢＩＵは、最高位アドレスと最低位アドレスを指定する２個のレジスタを備える。これらのアドレスは、ＰＣＩバスからＢＩＵの対応するセルのメモリに引き渡すことができる。これらのレジスタは、例えば、ＨＩＧＨ及びＬＯＷレジスタのように命名される。各ＢＩＵは、ＰＣＩバス上で行われるバス・トランザクションを監視する。ここで図５を参照する。ステップ８０５及び８１０に示されるように、ＢＩＵはＨＩＧＨ及びＬＯＷレジスタによって指定される範囲内のアドレスに対するアクセス要求を検出すると、そのトランザクションに対して肯定応答を発行し、ステップ８１５で自セルのローカル・メモリにトランザクションを転送する。各セルのＢＩＵ内のＨＩＧＨ及びＬＯＷレジスタによって指定されるアドレス範囲は重複してはならない。ＨＩＧＨ及びＬＯＷレジスタ内に格納される値は、ブートアップ時にシステムによって初期化される。

【００５２】本発明は、さらに、各スレーブ・セル内にＤＭＡモジュールを提供する。このＤＭＡモジュールは、マスター・セルのメモリ（リモート・メモリ）とスレーブ・セルのメモリ（ローカル・メモリ）間でやり取りされるデータ転送要求を管理する。この転送を制御するための状態（すなわち、レジスタ）には数種類がある。これらのレジスタは、メモリ・マッピングによってアクセスされる。例えば以下のようなレジスタを備えることができる。

【００５３】ＳＴＡＴＵＳレジスタ：　この３２ビット・レジスタは、ＤＭＡが処理している転送の状況及びエラー情報を提供する。

【００５４】ＳＲＣレジスタ：　この３２ビット・レジスタは、マスター・メモリ・ブロックとの間で送受信されるローカル・アドレス・ブロックの開始アドレスを保持する。

【００５５】ＤＳＴレジスタ：　この３２ビット・レジスタは、転送中のローカル・メモリ・ブロックで上書きされるリモート・アドレス・ブロックの開始アドレスを保持する。

【００５６】ＬＥＮレジスタ：　この３２ビット・レジスタは、読み取り動作と書き込み動作によって振る舞いが異なる。このレジスタに対する書き込み動作では、ＳＲＣ及びＤＳＴを使用した転送が開始され、その後、書き込まれたデータが転送長を指定する。このレジスタからの読み取り動作では、現在実行中のデータ転送の残余長が捕捉される。

【００５７】これらのレジスタは、事前決定されたＰＣＩバスのアドレス・スペースにメモリ・マッピングされる。すなわち、各セルはＰＣＩバス上の該当するアドレ

スから読み取ったり、それに書き込んだりすることによってこれらのレジスタにアクセスできる。

【００５８】次に、本発明のシステムにおけるデータ転送動作について説明する。先に述べたように、ネットワーク通信におけるセル間の最も基本的な動作は送信動作である。この送信動作は、データ転送における発信セルと受信セルの組み合わせによって、（ｉ）マスターからスレーブの転送、（ｉｉ）スレーブからマスターの転送、及び（ｉｉｉ）スレーブ間の転送、の３種類に分類される。次に、これら３種類の送信動作について個別に説明する。

【００５９】マスターからスレーブへの転送

図６のフローチャートに示すように、マスター・メモリ内のデータ・ブロックが転送先スレーブＢＵＩ内のＤＭＡモジュールによって転送されると、以下のステップが続いて実行される。すなわち、ステップ９０５で、マスターＣＰＵが、未完了のトランザクションがあるかどうか判断するためにターゲット・スレーブ・セル内のＳＴＡＴＵＳレジスタを読み取る。ステップ９１０で未完了のトランザクションがあると判断された場合、ステップ９１５で現在のトランザクションが事前に決定された時間にわたってスリープ状態に入り、これが経過するとステップ９０５及び９１０が繰り返される。一方、ステップ９１０で得られたＳＴＡＴＵＳレジスタからの応答データが、ターゲット・セル内のＤＭＡが新たな転送を開始できる状態にあることを示している場合は、ステップ９２０でマスターＣＰＵが該当するアドレスを有するターゲット・セル内のＳＲＣレジスタに書き込む。その後、ステップ９２５でマスターＣＰＵはＳＲＣレジスタへの書き込みが成功したかどうか確認する。成功した場合、マスターＣＰＵはステップ９３０に進み、ターゲット・セル内のＬＥＮレジスタに書き込みを行って転送を開始させる。

【００６０】ＤＳＴレジスタは初期化時にスレーブＣＰＵによって設定されるため、この時点ではＤＳＴレジスタへの書き込みは行われない。ＳＴＡＴＵＳレジスタによってＤＭＡモジュールが新たな転送を開始できる状態にあることが示されている場合に、メモリ内の転送先データ・バッファが準備完了状態にあるかどうかを確認することはスレーブの責任となる。

【００６１】転送実行中、ステップ９３５で、ＤＭＡモジュールは転送の進行にともなってＬＥＮレジスタの値を一定間隔で減少させる。ＬＥＮレジスタが０となったら、ＤＭＡはステップ９４０でスレーブＣＰＵへの割り込みを生成する。この割り込みによって、転送が終了し、新たなデータ転送に備えてＤＭＡモジュールが再初期化される。

【００６２】スレーブからマスターへの転送

図７のフローチャートに示すように、スレーブ・メモリ内のデータ・ブロックがマスターのＣＰＵ（ＣＰＵ０）に転送されるデータ転送では、以下のステップが実行される。この転送は、ステップ１００５でスレーブＣＰＵがマスターＣＰＵに対する割り込みを生成することによって開始される。割り込みを受信すると、マスターＣＰＵはステップ１０１０でＰＣＩバス内のメモリ・エイリアス領域にアクセスすることによって、スレーブのメモリ・ブロックからデータを読み取る。その後、マスターＣＰＵはステップ１０１５でデータを自セルのメモリに書き込み、スレーブ・セルからマスター・セルへのデータ転送を完了させる。

【００６３】スレーブ間の転送

図８のフローチャートに示すように、転送元スレーブのメモリ内のデータ・ブロックが転送先スレーブのＣＰＵに転送される場合には以下のステップが実行される。ステップ１１０５で、転送元スレーブのＣＰＵは転送先スレーブのＣＰＵに対する割り込みを生成する。ステップ１１１０で、転送先スレーブのＣＰＵは、ＰＣＩバス内のメモリ・エイリアス領域にアクセスすることによって、転送元スレーブのメモリ・ブロックからデータを読み取る。その後、転送先スレーブのＣＰＵはステップ１１１５でデータを自セルのローカル・メモリに書き込み、スレーブ間データ転送を完了させる。

【００６４】本発明の第２の実施の形態では、共用バスではなく、２点間接続を使用するセル間接続方式が使用される。したがって、セル１５０は、例えば図９に示すような交換ファブリック１６１によって接続することができる。この交換ファブリック１６１の例としては、クロスバー・ネットワーク、多段交換ネットワーク、時分割共用バスなどが挙げられる。

【００６５】この接続方式のデータ転送トランザクションは、例えば、転送元アドレス（セル：　メモリ）、転送先アドレス（セル：　メモリ）、転送タイプ、及び実際のデータ、というデータ・アイテムによって定義できる。

【００６６】例えば、セル０～セル３のメッセージを送信する場合であれば、交換ファブリックを介して移動するトランザクションは以下のデータ値によって実行できる。

【００６７】

転送元アドレス：　；セル０：　０ｘ００ａ００００
転送先アドレス：　；セル３：　０ｘ０００４０００
タイプ：　；書き込み
データ：

【００６８】場合によっては、このデータ転送は共用バス通信方式を使用する実施の形態よりも簡単に実行できることがある。

【００６９】以上好ましい実施の形態及び実施例をあげて本発明を説明したが、本発明は必ずしも上記実施の形態及び実施例に限定されるものではなく、その技術的思想の範囲内において様々に変形して実施することができ

る。

【００７０】明細書に組み込まれその一部を構成する添付図面は、本発明の現在好適な実施の形態を図示し、上記の概略説明及び好適な実施の形態の詳細説明とともに、本発明の原理を説明するものである。

【００７１】

【発明の効果】以上説明したように本発明によれば、プロセッサがネットワーク・プロトコルを使用し、相互接続装置を介して互いに通信する環境が実現される。

【００７２】また、プロセッサがネットワーク・プロトコルを使用し、共用バスを介して互いに通信することができる環境が実現される。

【００７３】例えば、プロセッサがＴＣＰ／ＩＰプロトコルを使用し、共用ＰＣＩバスを介して互いに通信することが可能となる。

【００７４】さらに、プロセッサがネットワーク・プロトコルを使用し、交換相互接続装置を介して互いに通信することができる環境が実現される。

【図面の簡単な説明】

【図１】　本発明の好適な実施の形態によるシステム構成を示すブロック図である。

【図２】　本発明の代替の実施の形態によるシステム構成を示すブロック図である。

【図３】　マスター・セルのＢＩＵ（Ｂｕｓ　Ｉｎｔｅｒｆａｃｅ　Ｕｎｉｔ）によって実行される機能論理を示すフローチャートである。

【図４】　スレーブ・セルのＢＩＵによって実行される機能論理を示すフローチャートである。

【図５】　メモリ・エイリアス機能のステップを示すフローチャートである。

【図６】　マスター・セルからスレーブ・セルへのデータ転送におけるステップを示すフローチャートである。

【図７】　スレーブ・セルからマスター・セルへのデータ転送におけるステップを示すフローチャートである。

【図８】　スレーブ・セル間のデータ転送におけるステップを示すフローチャートである。

【図９】　本発明の他の好適な実施の形態によるシステム構成を示すブロック図である。

【図１０】　ネットワーク・プロトコル・スタックへのインターフェースを示す図である。

【図１１】　ネットワーク・プロトコルの機能層を説明するＯＳＩ参照モデルを示す図である。

【図１２】　典型的なインターネット・ブラウザー・アプリケーションのネットワーク・インターフェースを示す図である。

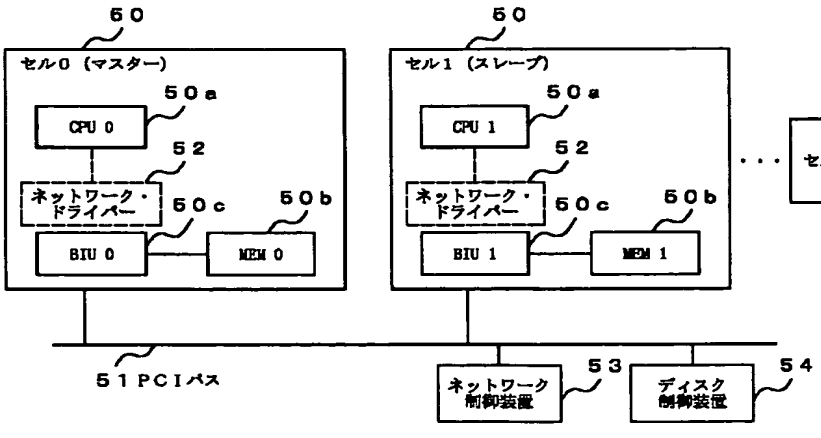
【図１３】　汎用相互接続装置ネットワークを示す図である。

【図１４】　分散メモリ・マルチプロセッサ・システムのアーキテクチャーを示す図である。

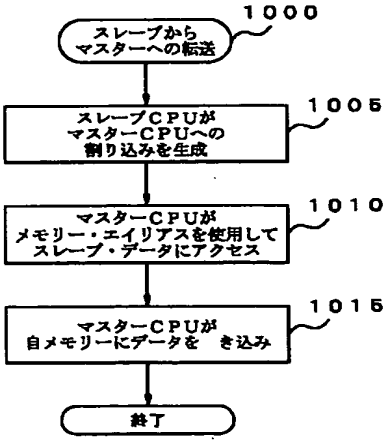
【符号の説明】

５０　セル
５０ａ　ＣＰＵ
５１　ＰＣＩバス
５２　ネットワーク・ドライバ
５３　ネットワーク制御装置
５４　ディスク制御装置
１５１ａ　ＰＣＩ制御装置
１５１　ＰＣＩバス
１５３　ネットワーク制御装置
１５４　ディスク制御装置
１５５　キャッシュ
１５６　ＣＰＵバス
１５７ａ　メモリ制御装置
１６１　交換ファブリック

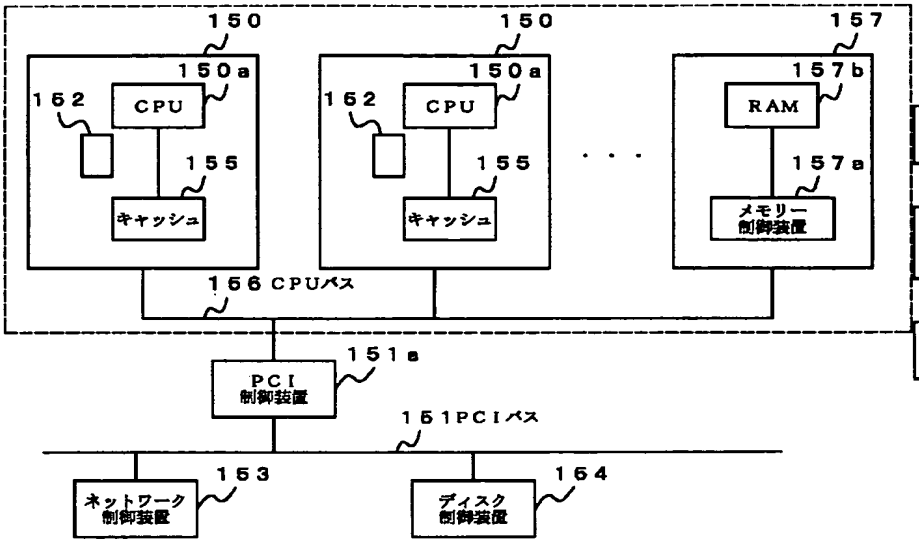
【図１】



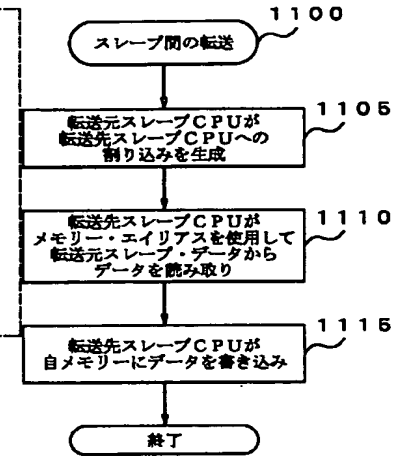
【図７】



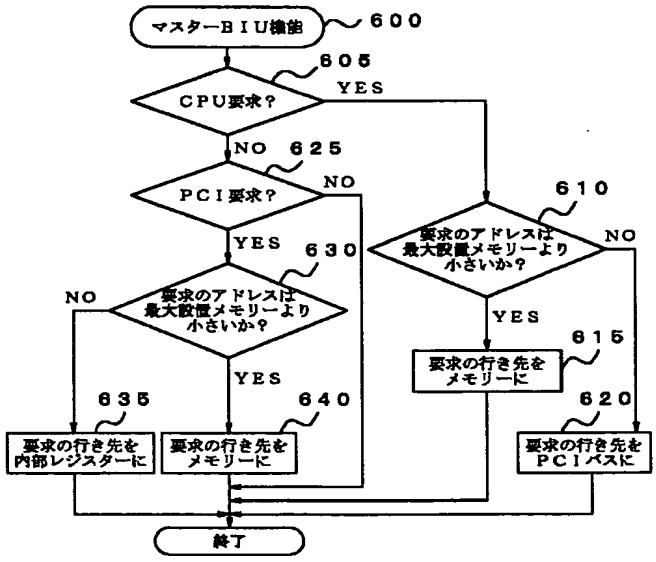
【図２】


150
150a
CPU
152
155
キャッシュ
150
150a
CPU
152
155
キャッシュ
157
157b
RAM
157a
メモリー制御装置
156 CPUバス
151a
PCI制御装置
151PCIバス
153
ネットワーク制御装置
154
ディスク制御装置

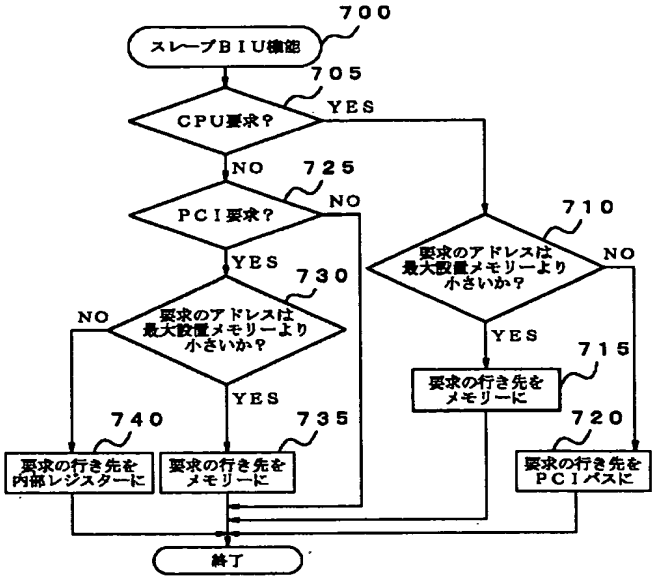
【図８】


1100
スレーブ間の転送
1105
転送元スレーブCPUが転送先スレーブCPUへの割り込みを生成
1110
転送先スレーブCPUがメモリー・エイリアスを使用して転送元スレーブ・データからデータを読み取り
1115
転送先スレーブCPUが自メモリーにデータを書き込み
終了

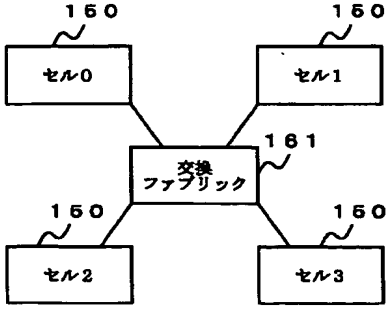
【図３】


マスターBIU機能
600
605
CPU要求?
YES
NO
625
PCI要求?
NO
YES
630
要求のアドレスは最大設置メモリーより小さいか?
NO
YES
610
要求のアドレスは最大設置メモリーより小さいか?
NO
YES
615
要求の行き先をメモリーに
620
要求の行き先をPCIバスに
635
要求の行き先を内部レジスターに
640
要求の行き先をメモリーに
終了

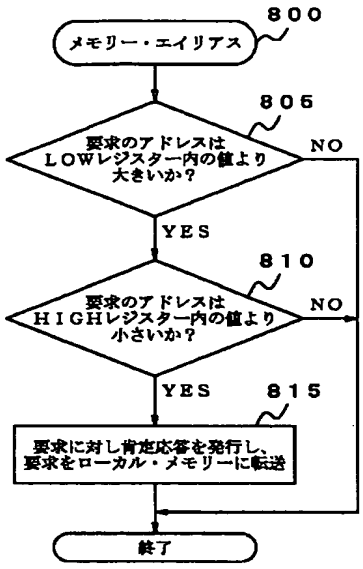
【図４】


700
スレーブBIU機能
705
CPU要求?
YES
NO
725
PCI要求?
NO
YES
730
要求のアドレスは最大設置メモリーより小さいか?
NO
YES
710
要求のアドレスは最大設置メモリーより小さいか?
NO
YES
715
要求の行き先をメモリーに
720
要求の行き先をPCIバスに
740
要求の行き先を内部レジスターに
735
要求の行き先をメモリーに
終了

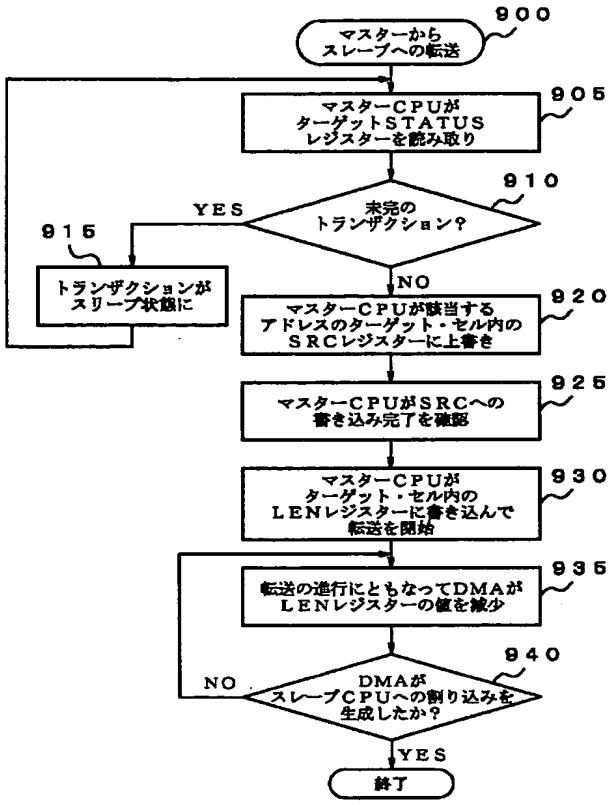
【図９】


150
セル0
150
セル1
161
交換ファブリック
150
セル2
150
セル3

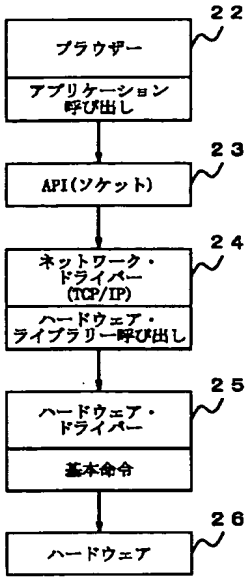
【図５】


800
メモリー・エイリアス
805
要求のアドレスは
ＬＯＷレジスター内の値より
大きいか？
NO
YES
810
要求のアドレスは
ＨＩＧＨレジスター内の値より
小さいか？
NO
YES
815
要求に対し肯定応答を発行し、
要求をローカル・メモリーに転送
終了

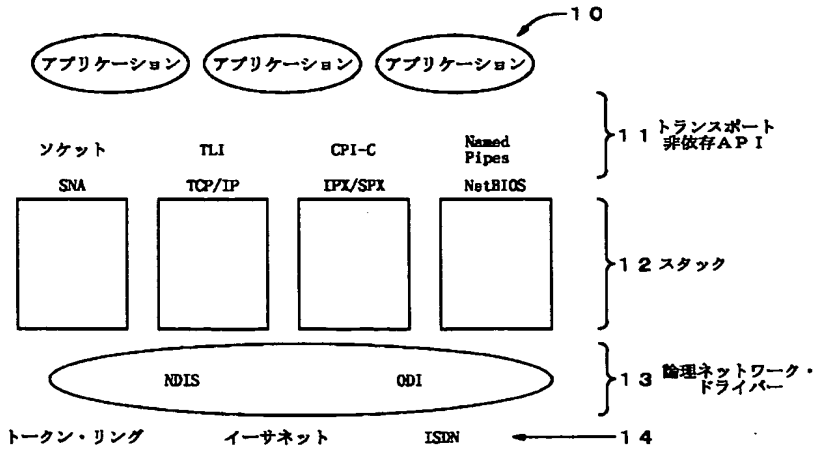
【図６】


900
マスターから
スレーブへの転送
905
マスターＣＰＵが
ターゲットＳＴＡＴＵＳ
レジスターを読み取り
910
未完の
トランザクション？
YES
915
トランザクションが
スリープ状態に
NO
920
マスターＣＰＵが該当する
アドレスのターゲット・セル内の
ＳＲＣレジスターに上書き
925
マスターＣＰＵがＳＲＣへの
書き込み完了を確認
930
マスターＣＰＵが
ターゲット・セル内の
ＬＥＮレジスターに書き込んで
転送を開始
935
転送の進行にともなってＤＭＡが
ＬＥＮレジスターの値を減少
940
NO
ＤＭＡが
スレーブＣＰＵへの割り込みを
生成したか？
YES
終了

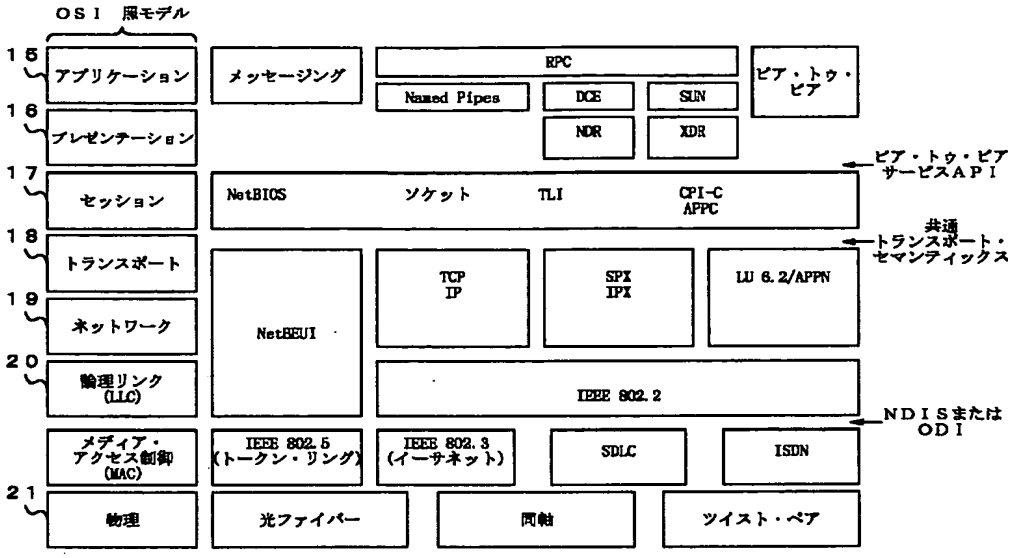
【図１２】


22
ブラウザー
アプリケーション
呼び出し
23
API(ソケット)
24
ネットワーク・
ドライバー
(TCP/IP)
ハードウェア・
ライブラリー呼び出し
25
ハードウェア・
ドライバー
基本命令
26
ハードウェア

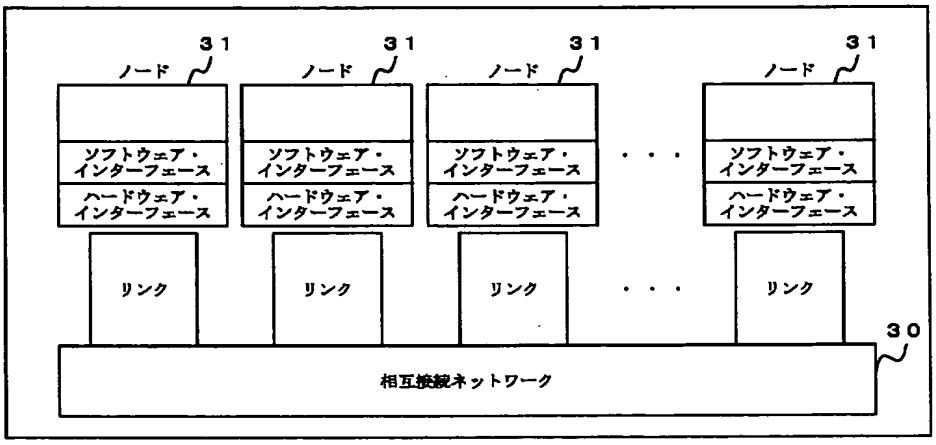
【図１０】


10
アプリケーション
アプリケーション
アプリケーション
ソケット
TLI
CPI-C
Named
Pipes
11 トランスポート
非依存ＡＰＩ
SNA
TCP/IP
IPX/SPX
NetBIOS
12 スタック
NDIS
ODI
13 論理ネットワーク・
ドライバー
トークン・リング
イーサネット
ISDN
14

【図１１】


ＯＳＩ 層モデル
１５
アプリケーション
１６
プレゼンテーション
１７
セッション
１８
トランスポート
１９
ネットワーク
２０
論理リンク (LLC)
メディア・アクセス制御 (MAC)
２１
物理
メッセージング
RPC
Named Pipes
DCE
SUN
NDR
XDR
ピア・トゥ・ピア
ピア・トゥ・ピア サービスＡＰＩ
NetBIOS
ソケット
TLI
CPI-C APPC
共通トランスポート・セマンティックス
NetBEUI
TCP IP
SPX IPX
LU 6.2/APPN
IEEE 802.2
NDISまたはODI
IEEE 802.5 (トークン・リング)
IEEE 802.3 (イーサネット)
SDLC
ISDN
光ファイバー
同軸
ツイスト・ペア

【図１３】


３１
ノード
ソフトウェア・インターフェース
ハードウェア・インターフェース
リンク
３０
相互接続ネットワーク

【図１４】